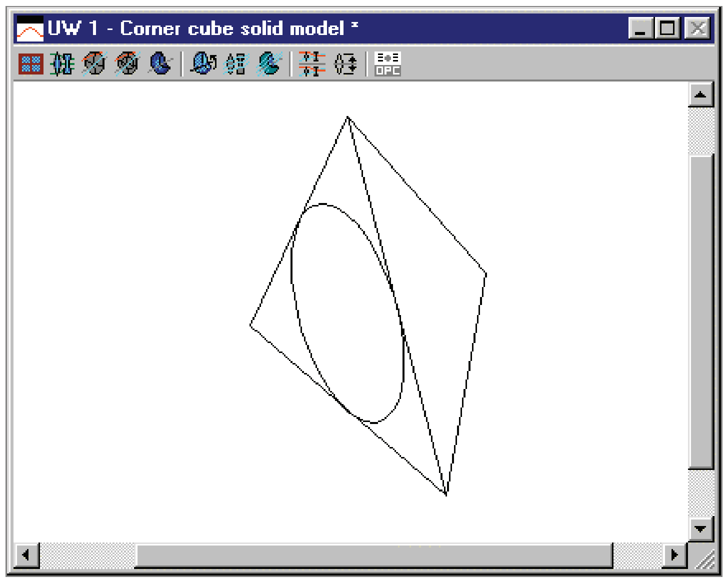
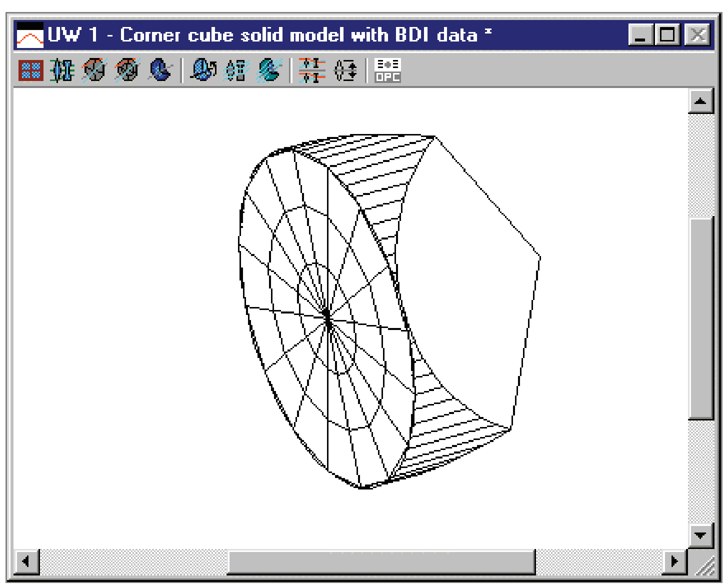
Corner-cube reflector

The corner-cube reflector, a combination of three plane mirrors at right angles to each other, has the property that rays entering the reflector exit from the reflector anti-parallel to the entering rays. A beam traversing a corner cube hits all three mirrors, but the order differs for different parts of the beam. Accordingly, corner cubes must be handled using non-sequential ray tracing. Generally, depending on the material and the conditions of use, a corner cube can function using total internal reflection, but there are reasons to "silver" the reflecting surface (discussed below), and the the standard corner cube prescription supplied in the OSLO demo library (cornercube.len) desginates the mirror surfaces as REFLECT.

The geometry of the corner cube reflector is such that rays entering the system perpendicular to the entrance face hit the first mirror at an angle of incidence of 54.73561 degrees. This makes the default drawing produced by OSLO look somewhat strange, as follows:



The portions of the entrance face outside the circular aperture are not optically useful, and typically when corner cubes are fabricated they are edged so that the fit into a cylindrical barrel. Accordingly, the cornercube.len model uses BDI data to produce 3D drawings (BDI data is ignored in plan-view drawings), as follows.



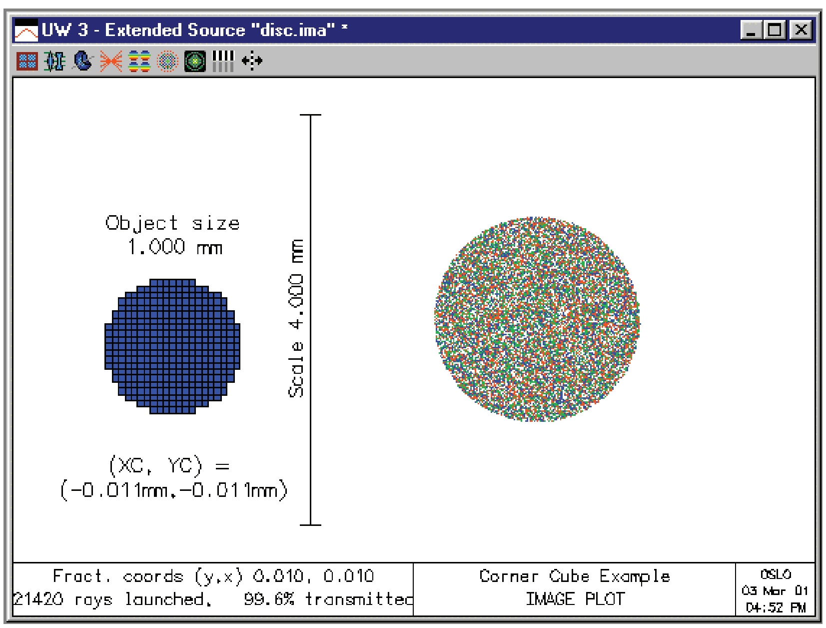
Unlike many BDI drawings, which use only a few vertices to define rectangular or triangular surfaces used in prisms, the data required for the corner cube model is quite extensive, because the cylindrical surface is simulated by facets. A few items of data are shown here to show that lines that define facets are drawn if the vertex number is positive, but supressed if it is negative. Thus in the following data for polygon facet 1, lines are drawn from vertex 4 to 1, and from vertex 1 to 2, but not from vertex 2 to 3. Regardless of whether the lines are drawn, however, the facet area is defined according to the PF specification, and hides the lines and facets behind it according to the viewing angle.

```
*BOUNDARY DRAWING DATA
 SRF 1:
 VX NBR         X            Y            Z         COORD SURF
    1          --           --          0.570000        1
    2        0.866025    -0.500000    -0.132893        1
    3        0.913545    -0.406737    -0.257040        1
    4        0.951057    -0.309017    -0.372080        1
    ...
 PF NBR        VX1          VX2          VX3          VX4
    1           1            2           -3           -1
    2          -1           -3           -4           -1
```

The specification of the non-sequential group for a corner-cube reflector is straightforward, and consists of the entrance port, the three reflecting surfaces, and the exit port. The other surfaces in the cornercube.len prescription are used to satisfy drawing requirements and are not optically active. Note that because of the complexity of the BDI data, it may be useful to insert the cornercube.len model in other systems as needed, using the Insert Lens file and Scale lens commands on the right-click spreadsheet menu.

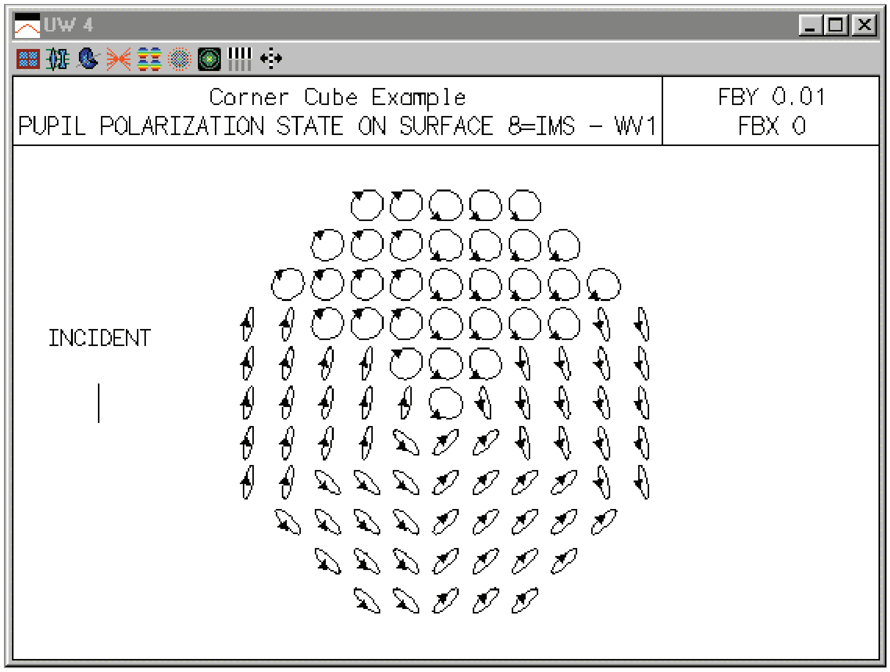
Historically, the corner-cube was one of the first applications for non-sequential ray tracing. According to optics folklore, a problem with double images was encountered when a particular optical system was being fabricated in the shop. The question arose as to whther these double images could be simulated using optical design software, so that the design could be adjusted. It turns out that the only design adjustment needed is to silver the reflecting faces, but as a pedigogical exercise, it is interesting to see what goes on when light retroreflects from a corner cube. To do this requires both non-sequential and polarization ray tracing, so it is necessary to use OSLO Premium.

If you illuminate a corner-cube with a small but finite incoherent source, a spot diagram, even a random one, will show even illumination, as shown below.



This is to be expected, since a spot diagram shows only the intersection points of rays with the image plane, and gives no information concerning the irradiance. Thus spot diagrams are useful for visualizing the shape of images formed by optical systems with aberration. In the case of a corner cube, however, there are no aberrations. On the other hand, there are polarization effects that can be readily observed using the polarization ray trace in OSLO.

To observe the polarization effects, you can use the cornercube.len model and turn on polarizatiion ray tracing in the General operating conditions. In addition, you must change the REFLECT designation on the mirror surfaces to AIR. If you do that, then you can use the Evaluate>>Polarization>>Pupil Polarization State command to produce the following plot.



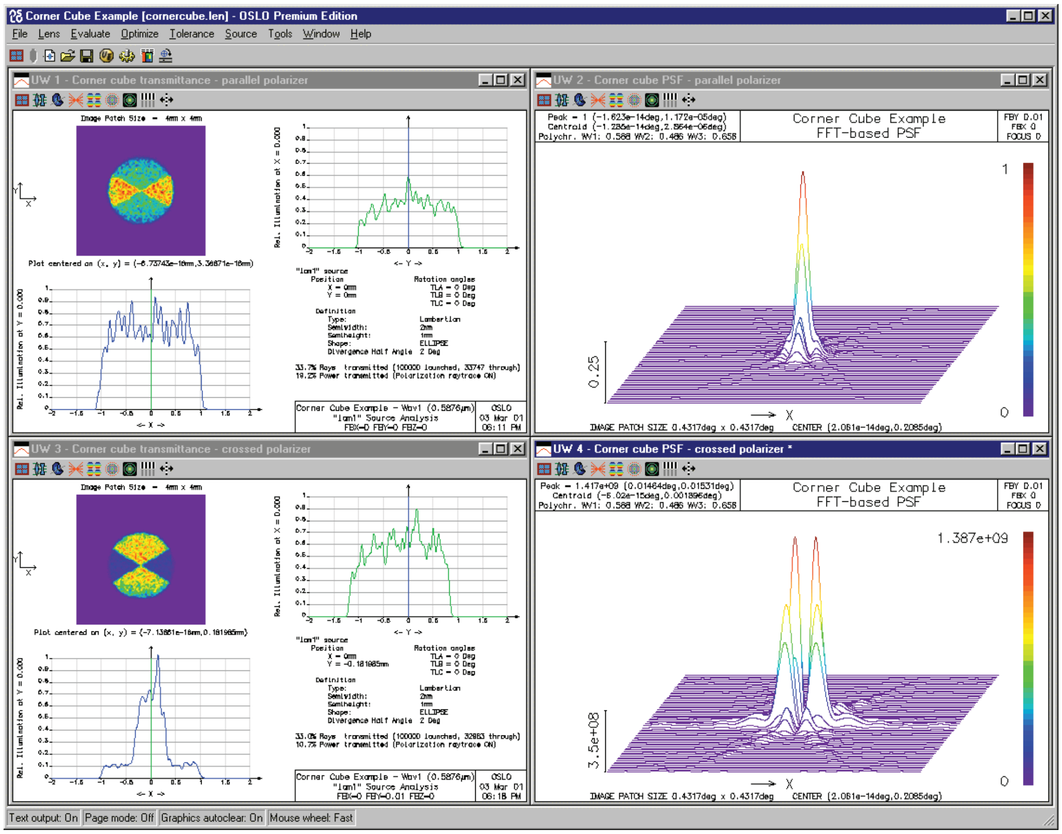
You see there are polarization effects that differ according to how light traverses the corner cube. Because of the unusual angle that rays intersect the mirror surfaces, different hexagonal sectors undergo different phase shifts that lead to the output polarization states shown. The arrowheads indicate the direction of rotation of the electric vector, showing that the output beam includes both right and left elliptically polarized components, when the incident beam is linearly polarized in the vertical direction.

You can analyze the system in more detail using the extended source and point spread function features in OSLO. The figure on the next page shows the results of such an analysis, produced by using a polarizing element on the output surface as follows, here shown as a crossed (i.e. *x*) polarizer. To set up a parallel polarizer, you use JA = 0.0 and JD = 1.0.

```
*POLARIZATION ELEMENT DATA
              AMPLITUDE     PHASE              AMPLITUDE     PHASE
  7      JA    1.000000      --          JB        --          --
         JC       --         --          JD        --          --
```

To produce the results on the following page, you need to set the object distance to 50mm for the illumination calculation, and 1e20 for the spread function calculation. In addition, you should set the object height slightly off axis (FBY = 0.01) to introduce a slight amount of aberration to prevent the crossed polarization case from blowing up during PSF normalization (the ideal PSF for this case is zero on axis).

You see that there is intensity modulation introduced by the output polarizers, but this is not sufficient to cause the PSF effects, which are caused by the polarization itself.


Corner Cube Example [cornercube.len] - OSLO Premium Edition
File Lens Evaluate Optimize Tolerance Source Tools Window Help
UW 1 - Corner cube transmittance - parallel polarizer
UW 2 - Corner cube PSF - parallel polarizer
Corner Cube Example
FFT-based PSF
UW 3 - Corner cube transmittance - crossed polarizer
UW 4 - Corner cube PSF - crossed polarizer *
1.387e+09
Text output: On
Page mode: Off
Graphics autoclear: On
Mouse wheel: Fast